

RTL CHALLENGE

DAY-18:- Verilog Code for Implementation of ADDERS Using Behavioural/
CONDITIONAL Statements.

Software used:- Xilinx Vivado(2023.1)

Half Adder using Conditional:

DESIGN CODE:-

```
// Design Name: Conditional
// Module Name: HalfAdderUsingConditional
// Project Name: RTL Challenge

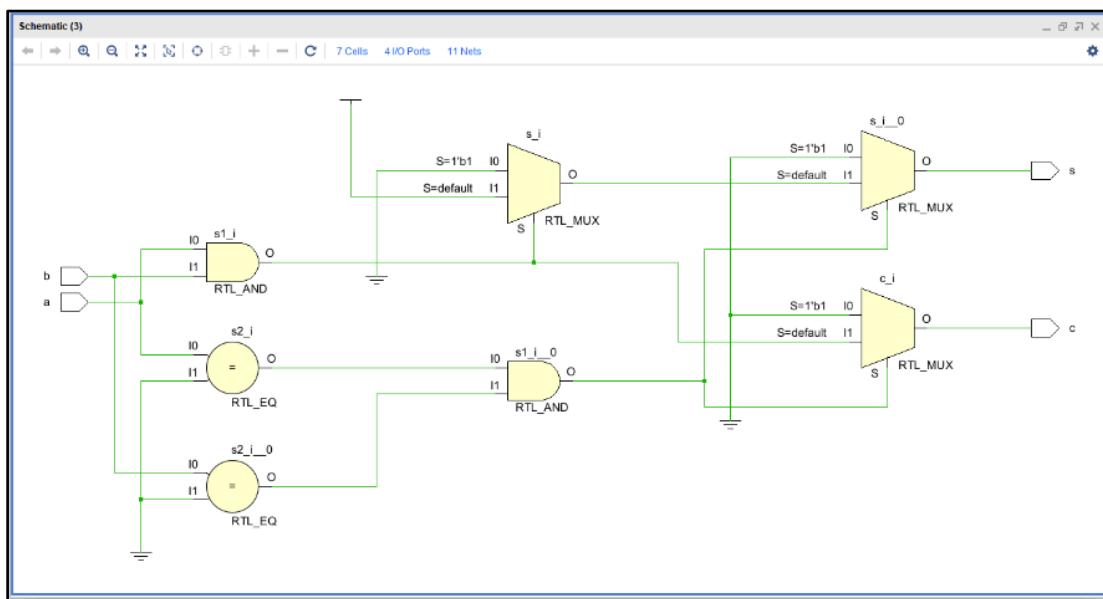
module HalfAdderUsingConditional(
    input a,b,
    output reg s,c
);
always@(a,b)
begin
    if(a==0&&b==0)
        begin
            s=0;
            c=0;
        end
    else if(a==1&&b==1)
        begin
            s=0;
            c=1;
        end
    else
        begin
            s=1;
            c=0;
        end
end
endmodule
```

TEST BENCH:-

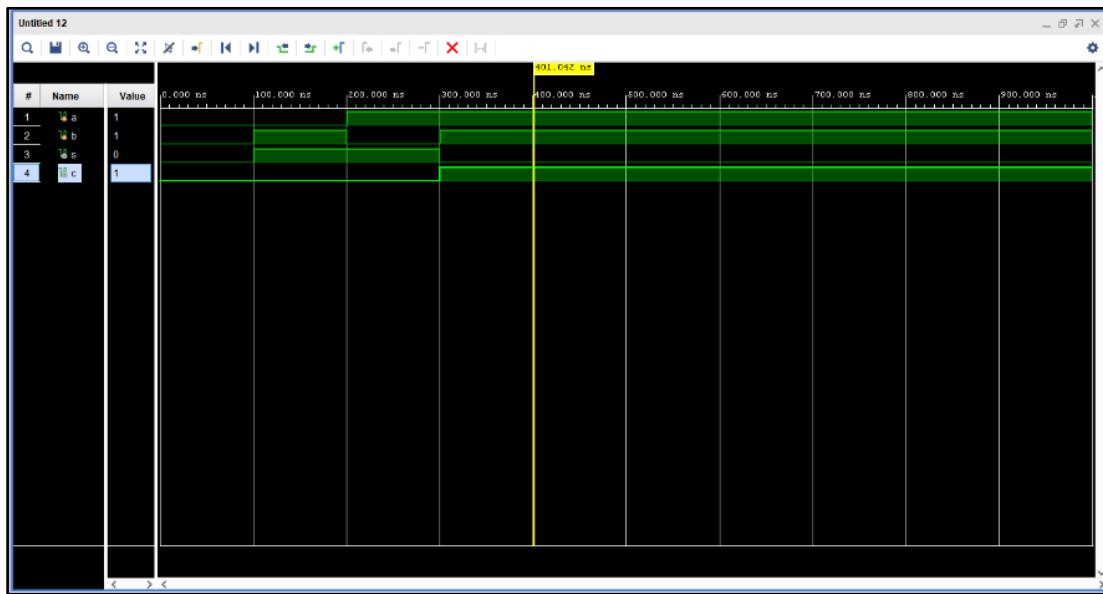
```
// Design Name: TEST BENCH
// Module Name: HalfAdderUsingConditional_tb
// Project Name: RTL DESIGN

module HalfAdderUsingConditional_tb();
reg a,b;
wire s,c;
HalfAdderUsingConditional uut (
    .a(a),
    .b(b),
    .s(s),
    .c(c)
);
initial
begin
    a=0;b=0;#100;
    a=0;b=1;#100;
    a=1;b=0;#100;
    a=1;b=1;#100;
end
endmodule
```

RTL SCHEMATIC:-



OUTPUT WAVE FORM:-



FULL ADDER USING CONDITIONAL:-

DESIGN CODE:-

```
// Design Name:  
// Module Name: FullAdderUsingConditional  
// Project Name:
```

```
module FullAdderUsingConditional(
```

```
    input a,b,cin,  
    output reg s,cout  
,
```

```
    always@(a,b,cin)  
    begin  
        if(a==0&&b==0&&cin==0)  
            begin  
                s=0;  
                cout=0;  
            end  
        end
```

```
else if(a==1&&b==1&&cin==1)
begin
    s=1;
    cout=1;
end

else if(a==0&&b==0&&cin==1)
begin
    s=1;
    cout=0;
end

else if(a==0&&b==1&&cin==0)
begin
    s=1;
    cout=0;
end

else if(a==1&&b==0&&cin==0)
begin
    s=1;
    cout=0;
end

else
begin
    s=0;
    cout=1;
end

end
```

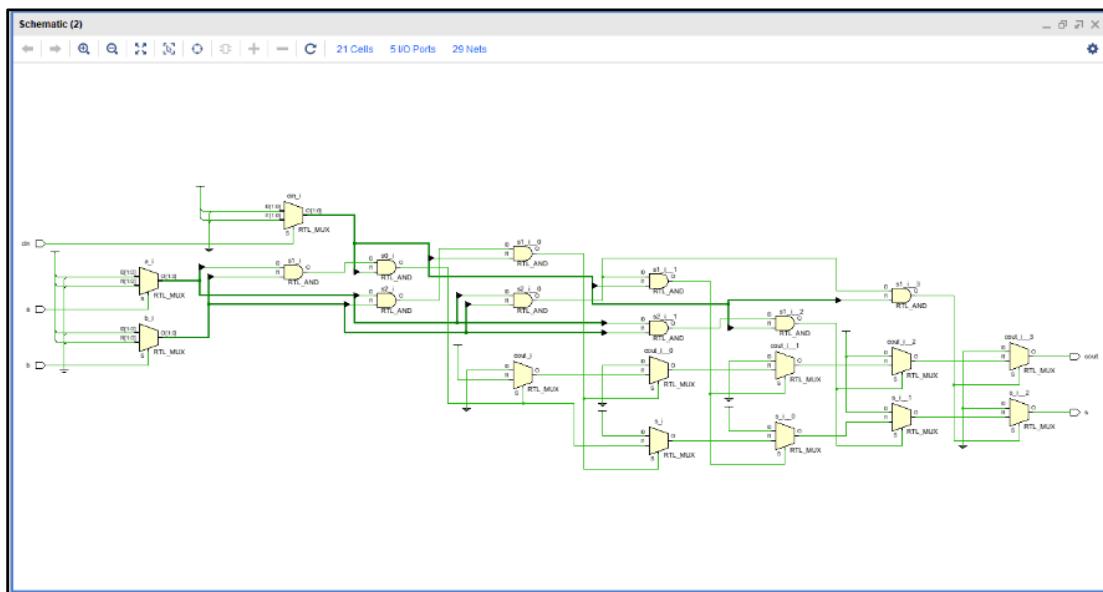
endmodule

TESTBENCH:

```
// Design Name: TEST BENCH
// Module Name: HalfAdderUsingConditional_tb
// Project Name: RTL DESIGN

module FullAdderUsingConditional_tb();
reg a,b,cin;
wire s,cout;
FullAdderUsingConditional uut (
    .a(a),
    .b(b),
    .cin(cin),
    .s(s),
    .cout( cout)
);
initial
begin
    a=0;b=0;cin=0;#100;
    a=0;b=0;cin=1;#100;
    a=0;b=1;cin=0;#100;
    a=0;b=1;cin=1;#100;
    a=1;b=0;cin=0;#100;
    a=1;b=0;cin=1;#100;
    a=1;b=1;cin=0;#100;
    a=1;b=1;cin=1;#100;
end
endmodule
```

RTL SCHEMATIC:



OUTPUT WAVEFORM:

